УНИВЕРСИТЕТ ИТМО

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
сентябрь–октябрь 2021 Том 21 № 5 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
September–October 2021 Vol. 21 No 5 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК УНИВЕРСИТЕТ ИТМО
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
сентябрь–октябрь 2021 Том 21 Номер 5

doi: 10.17586/2226-1494-2021-21-5-702-708

A meta-feature selection method based on the Auto-sklearn framework

Nikita I. Kulin[1]✉, **Sergey B. Muravyov**[2]

[1,2] ITMO University, Saint Petersburg, 197101, Russian Federation

[1] kylin98@list.ru✉, https://orcid.org/0000-0002-3952-6080
[2] smuravyov@itmo.ru, https://orcid.org/0000-0002-4251-1744

Abstract
In recent years, the task of selecting and tuning machine learning algorithms has been increasingly solved using automated frameworks. This is motivated by the fact that when dealing with large amounts of data, classical methods are not efficient in terms of time and quality. This paper discusses the Auto-sklearn framework as one of the best solutions for automated selection and tuning machine learning algorithms. The problem of Auto-sklearn 1.0 solution based on Bayesian optimization and meta-learning is investigated. A solution to this problem is presented. A new method of operation based on meta-database optimization is proposed. The essence of the method is to use the BIRCH clustering algorithm to separate datasets into different groups. The selection criteria are the silhouette measure and the minimum number of initial Bayesian optimization configurations. The next step uses a random forest model, which is trained on a set of meta-features and the resulting labels. Important meta-features are selected from the entire set. As a result, an optimal set of important meta-features is obtained, which is used to find the initial Bayesian optimization configurations. The described method significantly speeds up the search for the best machine learning algorithm for classification tasks. The experiments were conducted with datasets from OpenML to compare Auto-sklearn 1.0, 2.0 and a new version that uses the proposed method. According to the results of the experiment and statistical Wilcoxon T-criterion tests, the new method was able to outperform the original versions in terms of time, outperforms Auto-sklearn 1.0 and competes with Auto-sklearn 2.0. The proposed method will help to speed up the time to find the best solution for machine learning tasks. Optimization of such frameworks is reasonable in terms of saving time and other resources, especially when working with large amounts of data.

Keywords
AutoML, automated machine learning, machine learning, meta-learning, classification

For citation: Kulin N.I., Muravyov S.B. A meta-feature selection method based on the Auto-sklearn framework. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2021, vol. 21, no. 5, pp. 702–708. doi: 10.17586/2226-1494-2021-21-5-702-708

УДК 004.852

Метод отбора мета-признаков на основе фреймворка Auto-sklearn

Никита Игоревич Кулин[1]✉, **Сергей Борисович Муравьёв**[2]

[1,2] Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

[1] kylin98@list.ru✉, https://orcid.org/0000-0002-3952-6080
[2] smuravyov@itmo.ru, https://orcid.org/0000-0002-4251-1744

Аннотация
Предмет исследования. За последние годы задачу выбора и настройки алгоритмов машинного обучения все чаще решают с помощью автоматизированных фреймворков. Это мотивировано тем, что в работе с большим объемом данных классические методы не эффективны с точки зрения времени и качества. В работе рассмотрен фреймворк Auto-sklearn как одно из лучших решений для автоматизированного выбора и настройки алгоритмов машинного обучения. Исследована проблема решения Auto-sklearn 1.0, основанного на байесовской оптимизации и мета-обучении. Представлено решение данной проблемы. **Метод.** Предложен новый метод работы, основанный на оптимизации мета-базы. Сущность метода состоит в использовании алгоритма кластеризации BIRCH, для

разделения наборов данных по различным группам. Критериями отбора являются мера силуэта и минимальное количество начальных конфигураций байесовской оптимизации. На следующем шаге использована модель случайного леса, которая обучена на наборе мета-признаков и полученных метках. При этом отобраны важные мета-признаки из всего набора. В итоге получен оптимальный набор мета-признаков, который использован для нахождения начальных конфигураций байесовской оптимизации. **Основные результаты.** Рассмотренный метод позволяет значительно ускорить поиск лучшего алгоритма машинного обучения для задач классификации. Проведены эксперименты с наборами данных из OpenM, выполнено сравнение Auto-sklearn 1.0, 2.0 и новой версии, которая использует предлагаемый метод. По результатам эксперимента и статистических тестов на T-критерий Уилкоксона новый метод превосходит по времени оригинальные версии и по качеству Auto-sklearn 1.0 и составляет конкуренцию с Auto-sklearn 2.0. **Практическая значимость.** В результате метод позволяет уменьшить время нахождения лучшего решения для задач машинного обучения. Оптимизация таких фреймворков целесообразна с точки зрения экономии времени и других ресурсов особенно в работе с большим объемом данных.

Ключевые слова
AutoML, автоматическое машинное обучение, машинное обучение, мета-обучение, классификация

Ссылка для цитирования: Кулин Н.И., Муравьёв С.Б. Метод отбора мета-признаков на основе фреймворка Auto-sklearn // Научно-технический вестник информационных технологий, механики и оптики. 2021. Т. 21, № 5. С. 702–708 (на англ. яз.). doi: 10.17586/2226-1494-2021-21-5-702-708

Introduction

At the moment, one of the most common domains in machine learning is automatic machine learning (AutoML), which is in demand in many scientific and industrial fields [1, 2]. AutoML is used for the problem of finding the optimal algorithms in machine learning tasks; this problem is also known as Combined Selection and Hyperparameter Optimization (CASH). With the advent of the various AutoML frameworks, the machine learning problem-solving processes have been optimized in terms of time and efficiency. Finding the best "model-hyperparameter" configurations has become more objective and reasonable [3]. Today it is relevant in industrial environments where huge data clusters are used and searching for suboptimal configurations leads to various expenses [4].

Among frameworks, different approaches are used to find the optimal algorithm. For example, there are methods based on evolutionary algorithms, on Bayesian optimization; some of them use classical methods like Grid and Random Search. This paper deals with a classification task and uses the Auto-sklearn framework as a solution, which applies Bayesian optimization approach and meta-learning. The datasets for classification tasks were taken as experiments. The purpose of this paper is to demonstrate a new method of selection meta-features that is superior in finding efficient configurations compared to the original one.

Background

The task of finding the optimal algorithms can be classified into two subclasses: CASH (Combined Selection and Hyperparameter Optimization) and HPO (Hyperparameter Optimization). The difference between them is that HPO searches for the optimal configuration with respect to a single algorithm, while CASH considers optimization among several machine learning models and their hyperparameters. For this paper, we will be interested in CASH since it is more popular in industries.

One of the most popular solutions to this problem is Bayesian optimization. It is a global optimization method that builds a stochastic model of a mapping function from hyperparameter values to a target function applied on a test set. By iteratively applying a prospective hyperparameter configuration based on the current model and then updating it, Bayesian optimization seeks to gather as much information as possible about the function and the location of the optimum. One advantage of this approach is that it explores not only the neighborhood of the local error minima, but also other unexplored regions, thereby increasing the probability of finding new minimum locations. It has been shown experimentally that Bayesian optimization converges to an optimal solution faster than Grid and Random Search. So in the Data-Science community, Bayesian optimization has gained more demand and motivation in creating its various implementations, which we will describe below [5, 6].

Bayesian optimization

Sequential model-based optimization (SMBO) is one implementation of Bayesian optimization that finds a subspace or region of hyperparameters by which the algorithm will have a high metric value given the history of the considered hyperparameter points [7]. SMBO includes the following two parts: a probabilistic surrogate model and a selection function. The surrogate model is a model of the error function with respect to the hyperparameter location in the hyperparameter space. The essence is that we test several configurations in each iteration for their efficiency in solving the problem and update the surrogate model with the obtained results by recognizing better the neighborhood of the error minimum. The Gaussian processes (GP) and Tree-Structured Parzen Estimators (TPE) are used as a surrogate model. The preferred configurations are selected by a selection function, usually using the expected improvement.

SMBO was actively used for a long time, but at that time, it had a number of limitations. First, the space of hyperparameters was represented as continuous, hence it was impossible to embed categorical parameters, which were available in many ML models. Second, SMBO considered only one pipelining, which made it impossible to parallelize the process. These problems are solved by extending Sequential Model-based Algorithm Configuration (SMAC) with an ensemble of regression trees and adding several hyperparameters and algorithms. Based on experimental results, SMAC was able to outperform its competitors significantly.

Meta-learning

Meta-learning is an approach to determine the most appropriate algorithm for a particular task from a portfolio of algorithms [8–10]. It is one of the active areas of learning, which has been described in various articles and textbooks. The basic idea behind meta-learning is to optimize the process of finding the best algorithm choice by selecting a portfolio of proposed configurations. The concept is as follows: each data set can be represented as a group of meta-features, which can be the number of quantitative, categorical features, the number of classes, class entropy, the number of missing values, etc. Datasets can be described in the space of such meta-features and store configurations with the highest metric result. Such a model is called a meta-base. The intuition is as follows: the datasets that have similar meta-features have similar best configurations. Thus, when solving a particular machine learning problem for a particular dataset, one can find the nearest neighbors from the meta-base and take configurations from them, building a portfolio to use as a "warm start" at the beginning of the search for optimal algorithms. Such a collocation is significantly superior in comparison to the conventional SMAC, Tree Parzen Estimators (TPE), and naive algorithm that is shown in Fig. 1, where an average rank and functional evaluations are in conventional units [11].

Auto-sklearn

Auto-sklearn is one of the modern frameworks for algorithm selection and hyperparameter optimization, based on the scikit-learn library [12]. The framework has a large number of examples and good documentation, is available for classification and regression problems, contains many algorithms for data processing, categorical parameter coding and handling missing values to enumerate different combinations of pipelines. Auto-sklearn 1.0 uses Bayesian optimization along with the meta-learning discussed earlier, as well as automatic ensemble construction. Auto-sklearn is currently one of the most efficient and fast frameworks, and its working principle is shown in Fig. 2.

Despite the good performance of the first version, the developers of the framework were not convinced that meta-learning helps to achieve better and faster results. In their opinion, the use of some meta-features had no rational explanation, and good meta-features are expensive to compute them. This was one of the reasons to develop a new version that does not take meta-learning as a basis, namely, Auto-sklearn 2.0 that uses a unique portfolio of configurations at the beginning of the search for the best algorithm. Developer experiments showed the superiority of the second version that is shown in Fig. 3, where an average rank is in conventional units [13].

However, there was no detailed information about problematic meta-features. This lack of information prompted us to investigate the meta-features used in Auto-sklearn 1.0. While examining the framework and meta-features, we found some problems. First, Auto-sklearn developers do not use the classifier landmarks that are proposed in the SMAC model. Second, there are mutually exclusive, highly correlated, and anti-correlated meta-

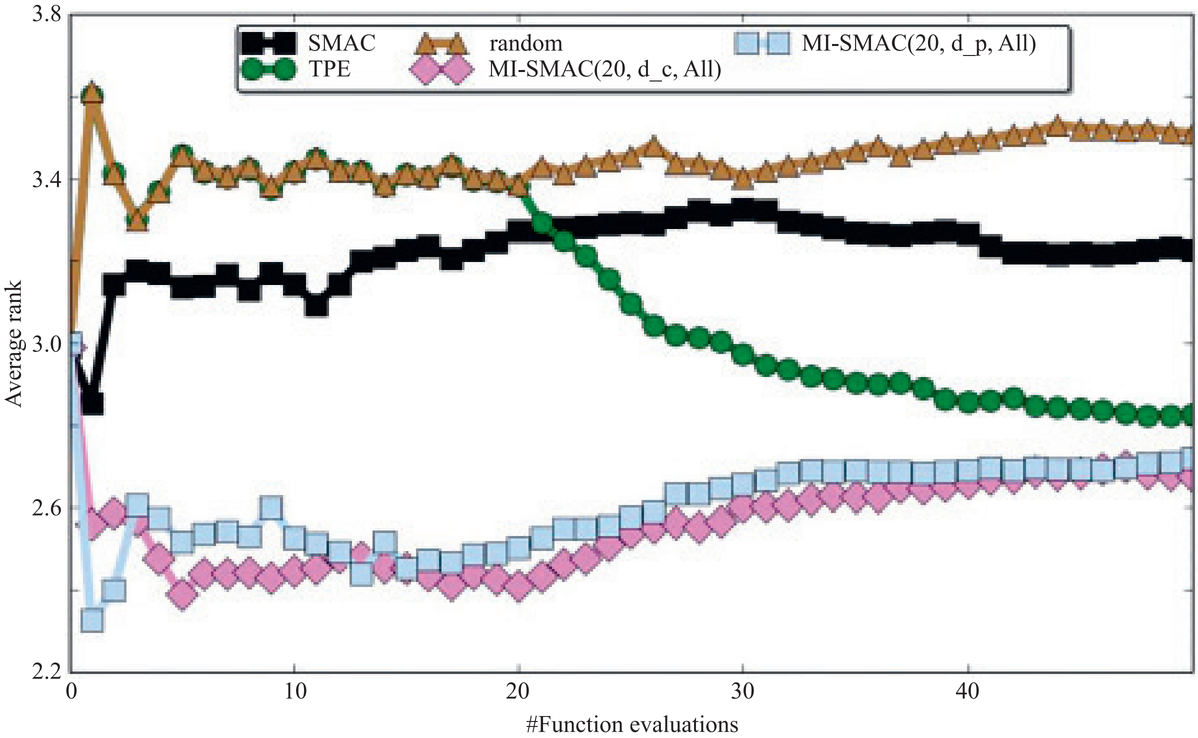

Fig. 1. Comparison of different search algorithms

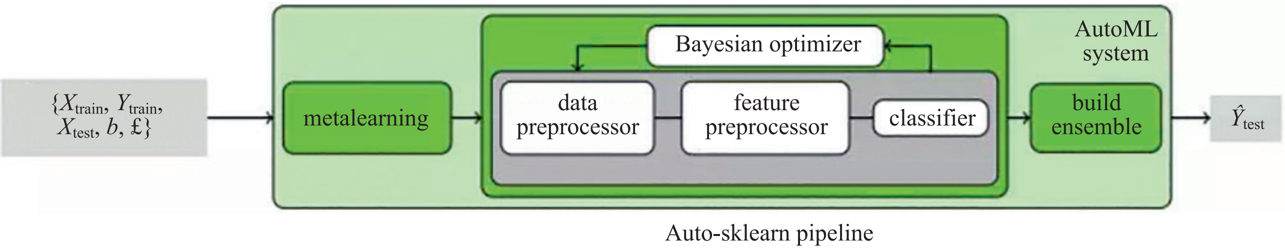

Fig. 2. Auto-sklearn 1.0 workflow

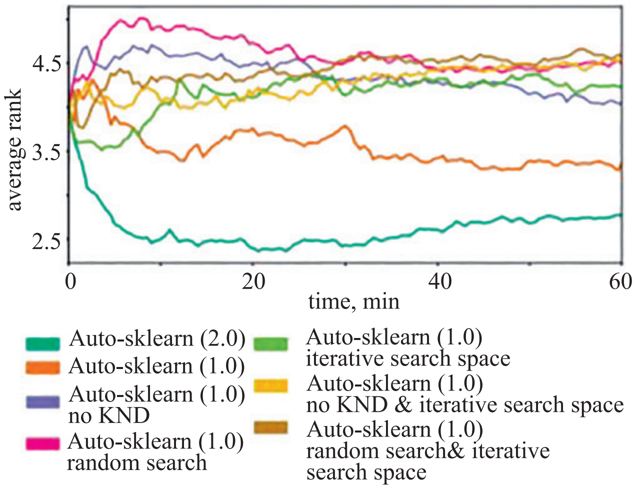

Fig. 3. Comparison of Auto-sklearn variations

features in the first version which are shown in Fig. 4. It leads to unnecessary computation and additional expansion of the meta-base, which is used in the nearest-neighbor search for a particular dataset and hence it slows down the search. Third, there are such meta-features that have their logarithmic counterparts.

We proposed the following hypothesis: the set of meta-features in Auto-sklearn 1.0 is not optimal, and the system may not achieve the fastest and best solution in the given time. In addition, because the meta-features were not studied, this may be the reason why Auto-sklearn 2.0 excelled in the experiment set by the developers, which made the first version uncompetitive. Our goal is to find a set of meta-features that produces the best results in many tasks in the fastest time.

Proposed method

In this paper, we propose to use a new method for selecting meta-features. The first original version of Auto-sklearn contains 40 meta-features. We had to look at all the different meta-features to find the optimal set. For this purpose, we used the Python meta-feature extractor (Pymfe) library, which has a large set consisting of several groups of meta-features: general, information-theoretic, model-based, orientational, relative landmarks, by clustering type, conceptual, by feature sets, and by complexity [14].

To evaluate which meta-features to use, we conducted the following procedure. First, the entire meta-base was replaced by rewriting the old meta-traits. For the new meta-base, the data sets from OpenML and Kaggle were taken. All the default meta-features from the Pymfe, about 225 meta-recognitions, were written, which were used to describe the datasets. The next step was to find meta-features that were good at separating datasets into groups of similar features. This could be an advantage for finding new configurations across the entire meta-base, as well as across a single group of attributes in order to reduce search time. To cluster the data in the new database, the following methods were required: *K*-means, agglomerative hierarchical clustering and BIRCH [15]. The quality of clustering was measured by the silhouette metric. The number of clusters should not exceed five, as the minimum number of initial configurations was required for one group of data, i.e. 25. As a result of the search for the best clustering algorithm with Optuna, the BIRCH method performed best, dividing the space into three groups according to the maximum achieved value of the silhouette metric – 0.113.

The next step is the direct selection of meta-features. For this purpose, the random forest algorithm was used, since it supports the feature importance function. To set up the hyperparameters of the random forest, Optuna has also been used. As a result, the following random forest hyperparameters were configured: the tree depth is equal to 15, and the number of trees is equal to 864. A cross-validation showed 0.95 for accuracy metrics and 0.95 for f1-measure, which implied the successful training of the classifier. The random forest was maintained using the Pickle to predict the class on the new data. The process of selecting meta-features according to their importance using the forest model was performed. The results showed that most of the meta-features had an importance of no more than one percent, which meant that they had little influence on the classifier's decision to assign a group type to datasets. Therefore, meta-features with less than one percent importance were excluded. As a result, 24 meta-features were selected for the new meta-base.

Comparison

For the current comparison, we proposed the AutoSklearn 1.2 version using the new meta-feature selection method, the original versions AutoSklearn 1.0 and 2.0, and the version without redundant and correlated meta-features Auto-sklearn 1.1. The experiment used 80 datasets, and the accuracy and f1 metrics are shown in conventional units. The results are shown in Fig. 5–6.

During the experiment, statistical Wilcoxon T-tests were conducted to accurately interpret the results described in Table, in which the proposed version of Auto-sklearn 1.2

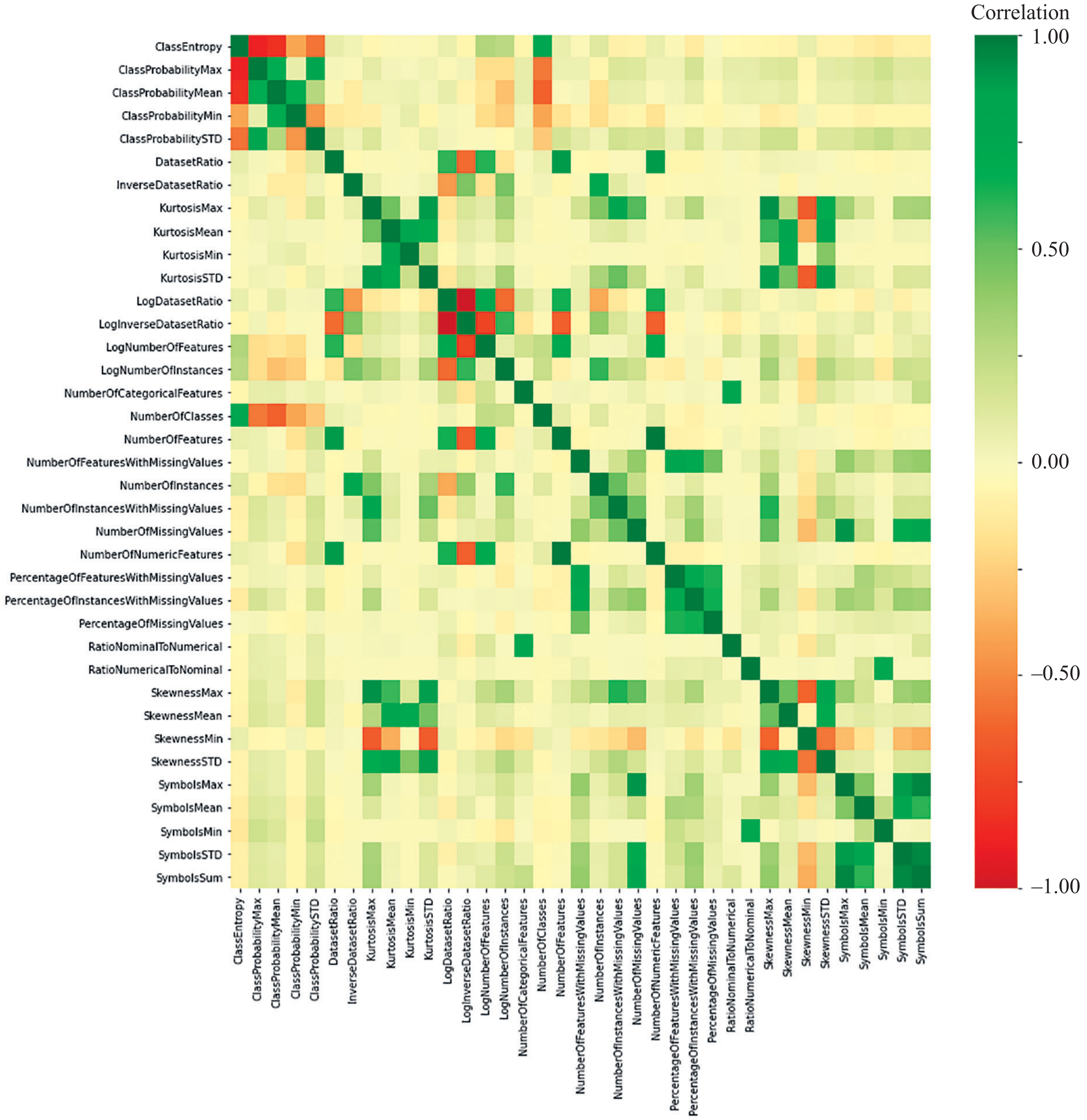

Fig. 4. Meta-feature Pearson correlation heatmap

showed superiority in quality measure over Auto-sklearn 1.0 and was competitive with Auto-sklearn 2.0, where F($x_{1.0}$), F($x_{1.1}$), F($x_{1.2}$), F($x_{2.0}$) are the location of Auto-sklearn version sets for f1 measures. In terms of speed, Auto-sklearn 1.2 was the fastest, with a time of 4 hours and 19 minutes, followed by Auto-sklearn 1.1 at 4 hours

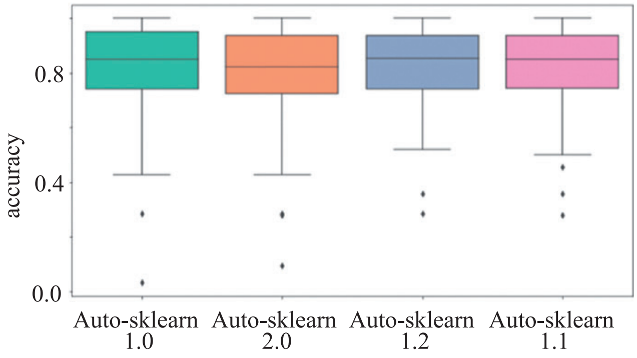

Fig. 5. Comparison between AutoSklearn 1.0, 2.0, and 1.2 on accuracy

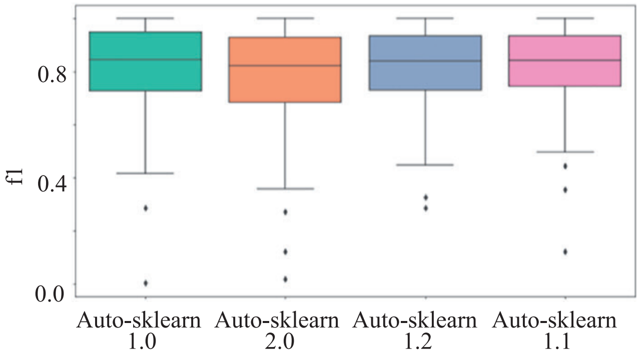

Fig. 6. Comparison between AutoSklearn 1.0, 1.1, 2.0, and 1.2 on f1

Table. Results of statistical tests according to the Wilcoxon test

H0: Null hypothesis	H1: Alternative hypothesis	T-test Wilcoxon	P-value
$F(x_{1.2}) = F(x_{1.0})$ f1	$F(x_{1.2}) > F(x_{1.0})$	585	0.047
$F(x_{1.2}) = F(x_{2.0})$ f1	$F(x_{1.2}) > F(x_{2.0})$	1177	0.572
$F(x_{2.0}) = F(x_{1.0})$ f1	$F(x_{2.0}) > F(x_{1.0})$	1335	0.071
$F(x_{1.1}) = F(x_{1.2})$ f1	$F(x_{1.1}) > F(x_{1.2})$	1528	0.600

and 20 minutes, Auto-sklearn 1.0 required 4 hours and 21 minutes, the second version was 4 hours and 25 minutes.

Conclusion

This paper proposes a new method for meta-feature selection using Auto-sklearn improvement as an example. We conducted the experimental studies and made a comparative analysis of the performance of the developed Auto-sklearn 1.2 system, which showed its superiority over Auto-sklearn 1.0 and competitiveness with Auto-sklearn 2.0 in terms of quality and slight superiority over both versions in terms of running time.

In the future, a number of works are planned on the system: expanding the meta-database to three hundred datasets, performing additional version comparisons, optimizing the calculation of meta-features, and refining the idea of a split meta-space.

References

1. Nagarajah T., Guhanathan P. A Review on Automated Machine Learning (AutoML) Systems. *Proc. IEEE 5th International Conference for Convergence in Technology (I2CT)*, 2019, pp. 9033810. https://doi.org/10.1109/I2CT45611.2019.9033810
2. Ge P. Analysis on approaches and structures of automated machine learning frameworks. *Proc. 2020 International Conference on Communications, Information System and Computer Engineering (CISCE)*, 2020, pp. 474–477. https://doi.org/10.1109/CISCE50729.2020.00106
3. Chauhan K., Jani S., Thakkar D., Dave R., Bhatia J., Tanwar S., Obaidat M.S. Automated machine learning: The new wave of machine learning. *Proc. 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2020, pp. 205–212. https://doi.org/10.1109/ICIMIA48430.2020.9074859
4. Ebadi A., Gauthier Y., Tremblay S., Paul P. How can automated machine learning help business data science teams? *Proc. 18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2019, pp. 1186–1191. https://doi.org/10.1109/ICMLA.2019.00196
5. Snoek J., Larochelle H., Adams R.P. Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 2012, vol. 4, pp. 2951–2959.
6. Jiang M., Chen Y. Research on bayesian optimization algorithm selection strategy. *Proc. IEEE International Conference on Information and Automation (ICIA)*, 2010, pp. 2424–2427. https://doi.org/10.1109/ICINFA.2010.5512281
7. Feurer M., Hutter F. Hyperparameter optimization. *Automated Machine Learning. Springer*, 2019, pp. 3–33. https://doi.org/10.1007/978-3-030-05318-5_1
8. Brazdil P., Giraud Carrier C., Soares C., Vilalta R. *Metalearning: Applications to Data Mining.* Springer Science & Business Media, 2009, XI, 176 p. https://doi.org/10.1007/978-3-540-73263-1
9. Hospedales T.M., Antoniou A., Micaelli P., Storkey A.J. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, in press. https://doi.org/10.1109/TPAMI.2021.3079209
10. Abdulrhaman S.M., Brazdil P. Measures for combining accuracy and time for meta-learning. *CEUR Workshop Proceedings*, 2014, vol. 1201, pp. 49–50.
11. Feurer M., Springenberg J., Hutter F. Initializing bayesian hyperparameter optimization via meta-learning. *Proc. 29th AAAI Conference on Artificial Intelligence*, 2015, pp. 1128–1135.
12. Feurer M., Klein A., Eggensperger K., Springenberg J.T., Blum M., Hutter F. Auto-sklearn: efficient and robust automated machine learning. *Automated Machine Learning*. Springer, 2019, pp. 113–134. https://doi.org/10.1007/978-3-030-05318-5_6
13. Feurer M., Eggensperger K., Falkner S., Lindauer M., Hutter F. Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning. *arXiv.org*. arXiv:2007.04074. 2020.

Литература

1. Nagarajah T., Guhanathan P. A Review on automated machine learning (AutoML) systems // Proc. IEEE 5th International Conference for Convergence in Technology (I2CT). 2019. P. 9033810. https://doi.org/10.1109/I2CT45611.2019.9033810
2. Ge P. Analysis on approaches and structures of automated machine learning frameworks // Proc. 2020 International Conference on Communications, Information System and Computer Engineering (CISCE). 2020. P. 474–477. https://doi.org/10.1109/CISCE50729.2020.00106
3. Chauhan K., Jani S., Thakkar D., Dave R., Bhatia J., Tanwar S., Obaidat M.S. Automated machine learning: The new wave of machine learning // Proc. 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). 2020. P. 205–212. https://doi.org/10.1109/ICIMIA48430.2020.9074859
4. Ebadi A., Gauthier Y., Tremblay S., Paul P. How can automated machine learning help business data science teams? // Proc. 18th IEEE International Conference on Machine Learning and Applications (ICMLA). 2019. P. 1186–1191. https://doi.org/10.1109/ICMLA.2019.00196
5. Snoek J., Larochelle H., Adams R.P. Practical bayesian optimization of machine learning algorithms // Advances in Neural Information Processing Systems. 2012. V. 4. P. 2951–2959.
6. Jiang M., Chen Y. Research on bayesian optimization algorithm selection strategy // Proc. IEEE International Conference on Information and Automation (ICIA). 2010. P. 2424–2427. https://doi.org/10.1109/ICINFA.2010.5512281
7. Feurer M., Hutter F. Hyperparameter optimization // Automated Machine Learning. Springer, 2019. P. 3–33. https://doi.org/10.1007/978-3-030-05318-5_1
8. Brazdil P., Giraud Carrier C., Soares C., Vilalta R. Metalearning: Applications to Data Mining. Springer Science & Business Media, 2009. XI, 176 p. https://doi.org/10.1007/978-3-540-73263-1
9. Hospedales T.M., Antoniou A., Micaelli P., Storkey A.J. Meta-learning in neural networks: A survey // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021. in press. https://doi.org/10.1109/TPAMI.2021.3079209
10. Abdulrhaman S.M., Brazdil P. Measures for combining accuracy and time for meta-learning // CEUR Workshop Proceedings. 2014. V. 1201. P. 49–50.
11. Feurer M., Springenberg J., Hutter F. Initializing bayesian hyperparameter optimization via meta-learning // Proc. 29th AAAI Conference on Artificial Intelligence. 2015. P. 1128–1135.
12. Feurer M., Klein A., Eggensperger K., Springenberg J.T., Blum M., Hutter F. Auto-sklearn: efficient and robust automated machine learning // Automated Machine Learning. Springer, 2019. P. 113–134. https://doi.org/10.1007/978-3-030-05318-5_6
13. Feurer M., Eggensperger K., Falkner S., Lindauer M., Hutter F. Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning // arXiv.org. arXiv:2007.04074. 2020.

14. Alcobaça E., Siqueira F., Rivolli A., Garcia L.P.F., Oliva J.T., de Carvalho A.C.P.L.F. MFE: Towards reproducible meta-feature extraction. *Journal of Machine Learning Research*, 2020, vol. 21, pp. 1–5.
15. Zhang T., Ramakrishnan R., Livny M. BIRCH: an efficient data clustering method for very large databases. *ACM Sigmod Record*, 1996, vol. 25, no. 2, pp. 103–114. https://doi.org/10.1145/235968.233324

Authors

Nikita I. Kulin — Student, ITMO University, Saint Petersburg, 197101, Russian Federation, SC 57222386134, https://orcid.org/0000-0002-3952-6080, kylin98@list.ru

Sergey B. Muravyov — PhD, Assistant, ITMO University, Saint Petersburg, 197101, Russian Federation, SC 57194035005, https://orcid.org/0000-0002-4251-1744, smuravyov@itmo.ru

Received 14.06.2021
Approved after reviewing 09.08.2021
Accepted 27.09.2021

14. Alcobaça E., Siqueira F., Rivolli A., Garcia L.P.F., Oliva J.T., de Carvalho A.C.P.L.F. MFE: Towards reproducible meta-feature extraction // Journal of Machine Learning Research. 2020. V. 21. P. 1–5.
15. Zhang T., Ramakrishnan R., Livny M. BIRCH: an efficient data clustering method for very large databases // ACM Sigmod Record. 1996. V. 25. N 2. P. 103–114. https://doi.org/10.1145/235968.233324

Авторы

Кулин Никита Игоревич — студент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, SC 57222386134, https://orcid.org/0000-0002-3952-6080, kylin98@list.ru

Муравьёв Сергей Борисович — кандидат технических наук, ассистент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, SC 57194035005, https://orcid.org/0000-0002-4251-1744, smuravyov@itmo.ru

Статья поступила в редакцию 14.06.2021
Одобрена после рецензирования 09.08.2021
Принята к печати 27.09.2021